NNexus Installation and API

James Gardner

February 28, 2007

This document outlines the installation, dependencies, and API of NNexus.

NNexus (Noosphere Networked Entry eXtension and Unification System) is an abstraction and generalization of the automatic linking component of the Noosphere system, which is the platform of PlanetMath (planetmath.org), PlanetPhysics (planetphysics.org), and other Noosphere sites.

Users of NNexus will apply the following basic functionality to their corpus. When an entry is rendered for display (either at display time or during offline batch processing), the text is broken down into tokens and scanned for words that invoke concepts that have been defined already (in other entries). These words (or word tuples) are ultimately turned into hyperlinks to the corresponding entries in the output rendering. In addition, when the concepts are added to the collection (or the set of concept labels otherwise changes), entries containing potential invocation of these concept labels are *invalidated* using a special inverted index called the *invalidation index*. This forces these entries to go through link analysis themselves by or before the next time they are displayed.

This automatic system almost completely frees the author from having to “think about links.” It addresses the problems of both outgoing and incoming links, with respect to a new entry or new concepts.

1 Installation

This section outlines the installation of NNexus server.

1.1 Dependencies

NNexus was originally developed on Mac OSX with the following software installed:

- perl-5.8.8 with the following modules installed from CPAN:

  ```
  Cwd;
  DBI;
  Data::Dumper;
  Encode;
  Switch;
  Time::HiRes
  Unicode::String
  XML::SAX;
  XML::Simple;
  XML::Writer;
  ```

  To install any of these modules run `perl -MCPAN -e shell`. In the shell type `install Module::Name`. This should install the latest versions of the modules. Note: It is very important to have the latest versions of the XML related modules as all information exchange with NNexus is performed using a strict XML syntax. In order for NNexus to understand all the XML the latest versions of the XML parsers must be installed.

- MySQL version 5.0.22 and later are recommended for installation.

NNexus can be run using some older versions of perl and mysql, but this is not guaranteed and may be a headache to get working. We have successfully installed NNexus on a Debian system running Perl version 5.x and MySQL version 4.0, but it wasn't pretty.

1.2 Quick and Dirty Install Process

Follow the steps below to get NNexus running.

1. Pray!

2. Create the NNexus database and tables. Open up `mysql` as a user that has permissions to create a database. First create the nnexus database (Any of the values in the examples we provide can be changed by changing the NNexus configuration file -discussed later.)

   ```
   create database nnexus;
   use nnexus;
   ```

   Now create the tables.

   ```
   source /path/to/nnexus/database/nnexus_schema.sql
   ```

   We also recommend that you create a specific database user with access only to the NNexus database. To learn how to do this check out the documentation of MySQL.

3. It is now time to modify the NNexus configuration files. Copy `baseconf-example.xml` to `baseconf.xml`. The example looks like this:

   ```
   <!-- this is the baseconfig file for NNexus -->
   <config>
   <domains>
   <domain>
   <name>planetmath.org</name>
   <link>http://link.to/xml/config/file/withpriorities</link>
   <urltemplate>http://planetmath.org/?op=getobj&amp;from=objects&amp;id=</urltemplate>
   <defaultscheme>msc</defaultscheme>
   </domain>
   <domain>
   <name>wikipedia.org</name>
   <link>http://link.to/xml/config/file/withpriorities</link>
   <urltemplate>http://wikipedia.org/encyclopedia/</urltemplate>
   <defaultscheme>msc</defaultscheme>
   </domain>
   </domains>

   <!-- Database configuration -->
   <database>
   <dbms>mysql</dbms>
   <dbname>nnexus</dbname>
   <dbuser>nnexus</dbuser>
   <dbpass>nnexus</dbpass>
   <dbhost>localhost</dbhost>
   </database>
   </config>
   ```

Modify the dbname, dbuser, dbpass, and dbhost to match the configuration of your MySQL server before you forget.

You can now modify the domain configuration. We now present a description of the XML fields.

- name - This just gives a name to the domain. All the domains must have different names.
- link - This is not really used yet. This will be supported in the future to allow for priorities for linking between domains and possibly some sort of domain specific linking guidance.
- urltemplate - Every document that is stored by NNexus is provided with a unique external (domain) identifier. When a link is made to an object the urltemplate is appended at the beginning to the external identifier.
- defaultscheme - This is the default classification scheme for the documents for the domain. When an object is added to NNexus it needs to be given a classification like scheme:3A123. If scheme is not provided when adding the object the defaultscheme is appended to the beginning of the class.

Now NNexus should be ready to run. Type `perl nnexuserver.pl` and let her rip. By default NNexus runs on port 7070 and we need to add an option for this in the config file.

2 API

This section describes how to communicate with the NNexus server.

All request to NNexus need to be sent using the following XML form:

```
<request>
...
</request>
```

where ... is any of the following requests. Note: The request must begin with a request tag followed by a newline and the end request must have a newline before and after.

2.1 Adding New Link Concepts

In order to add a new concept and link you may send:

```
<addobject>
<entry>
<title>same as above</title>
<defines>thing</defines>
<defines>widget</defines>
<synonym>term3</synonym>
<synonym>phrase of terms</synonym>
<domain>planetmath.org</domain>
<body>The body text</body>
<objid>a3db</objid>
<linkpolicy>permit 03A</linkpolicy>
<author>1</author>
<class>012A</class>
<class>02ADD</class>
</entry>
<entry>
... Same form as above.
</entry>
</addobject>
```

Notice that it is possible to add multiple objects with one request.

After adding (modifying) an entry to (in) Nnexus the system will return a list of object ids that need to be invalidated in the following form:

```
<response>
<invalid>ExternalID</invalid>
<invalid>AnotherExternalID</invalid>
</response>
```

We can delete an object by passing

```
<deleteobject>
<objid>1232</objid>
<domain>wikipedia.org</domain>
</deleteobject>
```

2.2 Getting invalid objects

A user may also request all invalidated objects by sending

```
<getinvalidobjects/>
```

and the server will return all invalidated objects as

```
<update>
<invalidate>
<objid>12</objid>
<objid>123</objid>
<objid>145</objid>
<objid>1125</objid>
</invalidate>
</update>
```

2.3 Obtaining Links for New Articles or Terms

There will be three schemes for obtaining links from Nnexus. The first two will use the entire article and the last will only find links for terms. Note: Only the Simple Mode is currently supported.

2.3.1 Simple Mode

```
<article> full text of article </article>
<class>03FA2</class>
```

this will return

```
<response>
<body>
full text of article with links added.
</body>
<links>
a list of the links in http://form separated by commas.
</links>
</response>
```

2.3.2 Detailed Mode

```
<detailedreq>
<article> full text of article </article>
<class>03FA2</class>
</detailedreq>
```

this will return

```
<detailedresp>
<article> full text of article </article>
<entry>
<place>12,13</place>
<option>
<link>link.to/article</link>
<domain>domain.com</domain>
</option>
<option>
<link>link.to/otherarticle</link>
<domain>otherdomain.com</domain>
</option>
</entry>
<entry>
<place>12,13</place>
<option>
<link>link.to/article</link>
<domain>domain.com</domain>
</option>
</entry>
</detailedresp>
```

2.3.3 Interactive Mode

```
<req>
<term>term or phrase</term>
<class>03FA2</class>
<dclass>desired classification of article to link to</dclass>
</req>
```

Note: dclass is optional After sending this information the server will return

```
<resp>
<link>http://link.to/the_article</link>
<linkid>12313</linkid>
</resp>
```

2.4 Exporting Concept Labels

To access the concept labels that Nnexus is currently using you may pass

```
<getconcepts>
<domain>domain.org</domain>
<domain>otherdomain.org</domain>
</getconcepts>
```

and the server will return the xml config files as

```
<config>
<domain>domain.org</domain>
<contents>contents of domain.org config file</contents>
</config>
<config>
<domain>otherdomain.org</domain>
<contents>contents of otherdomain.org config file</contents>
</config>
```

3 Examples